SYM-AM-16-049

PROCEEDINGS

OF THE

THIRTEENTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

THURSDAY SESSIONS VOLUME II

Blockmodeling and the Estimation of Evolutionary Architectural Growth in Major Defense Acquisition Programs

LTC Matthew Dabkowski, U.S. Army, University of Arizona
Ricardo Valerdi, Associate Professor, University of Arizona

Published April 30, 2016

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

Panel 12. Considerations for Focusing Development and Controlling Growth in MDAPs

Thursday, May 5, 2016	
9:30 a.m. – 11:00 a.m.	**Chair: Nancy Spruill,** Director, Acquisition Resources & Analysis, Office of the Under Secretary of Defense for Acquisition, Technology, & Logistics ***Blockmodeling and the Estimation of Evolutionary Architectural Growth in Major Defense Acquisition Programs*** LTC Matthew Dabkowski, U.S. Army, University of Arizona Ricardo Valerdi, Associate Professor, University of Arizona ***An Assessment of Early Competitive Prototyping for Major Defense Acquisition Programs*** William Fast, COL, U.S. Army (Ret.), Senior Lecturer, NPS ***The Sixth-Generation Quandary*** Raymond Franck, Professor Emeritus, U.S. Air Force Bernard Udis, Professor Emeritus, University of Colorado at Boulder

Blockmodeling and the Estimation of Evolutionary Architectural Growth in Major Defense Acquisition Programs[1]

LTC Matthew Dabkowski, USA—is a Lieutenant Colonel in the U.S. Army, and a PhD candidate in Systems and Industrial Engineering at the University of Arizona (UA). He received his BS in operations research from the United States Military Academy (USMA) in 1997, and his MS in systems engineering from the UA in 2007. LTC Dabkowski is the winner of the Military Operations Research Society's (MORS) 2013 Wayne P. Hughes Junior Analyst Award, the MORS 2012 Rist Prize, the TRADOC Analysis Center's 2012 LTC Paul J. Finken Memorial Award, and the Army's 2009 Dr. Wilbur B. Payne Memorial Award. [mfd1@email.arizona.edu]

Ricardo Valerdi—is an Associate Professor in the Systems and Industrial Engineering Department at the UA, where he teaches courses in cost estimation and systems engineering. He received his BS/BA in electrical engineering from the University of San Diego in 1999, and his MS and PhD in systems architecting and engineering from the University of Southern California in 2002 and 2005, respectively. Dr. Valerdi is the Editor-in-Chief of the *Journal of Cost Analysis and Parametrics*, as well as a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE). [rvalerdi@arizona.edu]

Abstract

It is well-known that cost overruns in Major Defense Acquisition Programs (MDAPs) are endemic, and requirements volatility is at least partially to blame. In particular, when the desired capabilities of a system change during its life cycle, substantial reengineering can result, especially when a new subsystem must be incorporated into an existing architecture. Of course, the likelihood and specifics of such additions are rarely known ahead of time, and predicting integration costs is challenging. In this paper, we present a novel algorithm to address this issue. In particular, leveraging an integer programming implementation of the social network analysis technique blockmodeling, we optimally partition the subsystems represented in Department of Defense Architecture Framework (DoDAF) models into architectural positions. Using this abstracted structure, we subsequently grow the architecture according to its statistical properties, and we estimate this unforeseen cost of evolutionary architectural growth via the Constructive Systems Engineering Cost Model (COSYSMO). We illustrate this process with a real-world example, discuss limitations, and highlight areas for future research.

[1] The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Introduction[2]

Major Defense Acquisition Programs (MDAPs) are notoriously prone to excessive cost overruns (GAO, 2011), and requirements volatility is often partially to blame (e.g., Bolten et al., 2008; Peña & Valerdi, 2015). In fact, based on the GAO's most recent *Assessments of Selected Weapon Programs* (2015), 6 of the 14 largest increases in MDAP development costs were due to the addition of new capabilities, making it the most frequent cause of substantial post-Milestone B (MS B) cost growth. Given a general lack of system specification early in the system life cycle (Blanchard & Fabrycky, 1998), this is not surprising, as accurately estimating the cost of an unknown set of capabilities is difficult at best.

With this in mind, in 2009, Congress passed the Weapon Systems Acquisition Reform Act (WSARA), which implemented several initiatives to rein in cost growth, including shifting an MDAP's baseline cost estimate from MS B to MS A (WSARA, 2009). Acknowledging the need for detailed system information earlier in the life cycle, the DoD followed suit in 2013 by requiring the submission of a draft Capability Development Document (CDD) pre–MS A (USD[AT&L], 2013), replete with the DoD Architecture Framework (DoDAF) models required by the *Joint Capabilities Integration and Development System* (JCIDS; Chairman of the Joint Chiefs of Staff, 2012).

Given WSARA's call for accurate early life cycle cost estimates, this has favorable implications. Specifically, in Valerdi, Dabkowski, and Dixit (2015), we demonstrate that the DoDAF models required pre–MS A map to 14 of the 18 parameters of the Constructive Systems Engineering Cost Model (COSYSMO). Consisting of four size drivers (i.e., number of requirements, number of interfaces, number of algorithms, and number of operational scenarios) and 14 effort multipliers, COSYSMO has been used by a variety of organizations to estimate the amount of systems engineering effort required to bring a system to fruition (e.g.,Valerdi, 2008; Wang et al., 2012),[3] and industry has found this estimate to be a valuable proxy for total system cost (e.g., Honour, 2004; Cole, 2012).

Moreover, in Dabkowski, Valerdi, and Farr (2014), we develop an algorithm to estimate the cost of unforeseen architectural growth in MDAPs via the SV-3 (or Systems-Systems Matrix), providing a mechanism to assess the cost risk associated with alternative designs. Leveraging elements of network science and simulation, the algorithm exploits both the micro- and macrostructure of the SV-3 to connect a new subsystem to an MDAP's existing architecture, and it employs COSYSMO to estimate the cost of the associated growth. In 2016, we validated and further refined our approach using real-world SV-3s (Dabkowski & Valerdi, 2016). While the details of our most recent work are beyond the scope of this paper, one of our modeling considerations is not, namely, the detection and exploitation of architectural communities within the SV-3.

---

[2] The material in the Introduction and Identifying and Exploiting Architectural Communities sections is derived from our earlier Acquisition Research Symposium paper titled "The Budding SV3: Estimating the Cost of Architectural Growth Early in the Life Cycle" (Dabkowski & Valerdi, 2014). Copyright is retained by the authors.

[3] COSYSMO estimates systems engineering effort in person months (nominal schedule) or PM_{NS}

Identifying and Exploiting Architectural Communities

In order to facilitate the discussion that follows, consider the hypothetical SV-3 in Panel (a) of Figure 1, where cell (i, j) is shaded if subsystem i interfaces with subsystem j, and darker shades indicate greater interface complexity (i.e., light gray ⇒ easy, medium gray ⇒ nominal, black ⇒ difficult). Consisting of N=20 subsystems (labeled A through T) and E=47 undirected interfaces,[4] suppose we are interested in estimating the effort required to incorporate an additional subsystem (U) into the architecture *without knowing its purpose or function.* In light of COSYMO's cost estimating relationship (CER), this ultimately forces us to estimate the number of interfaces (by complexity level) U will generate.

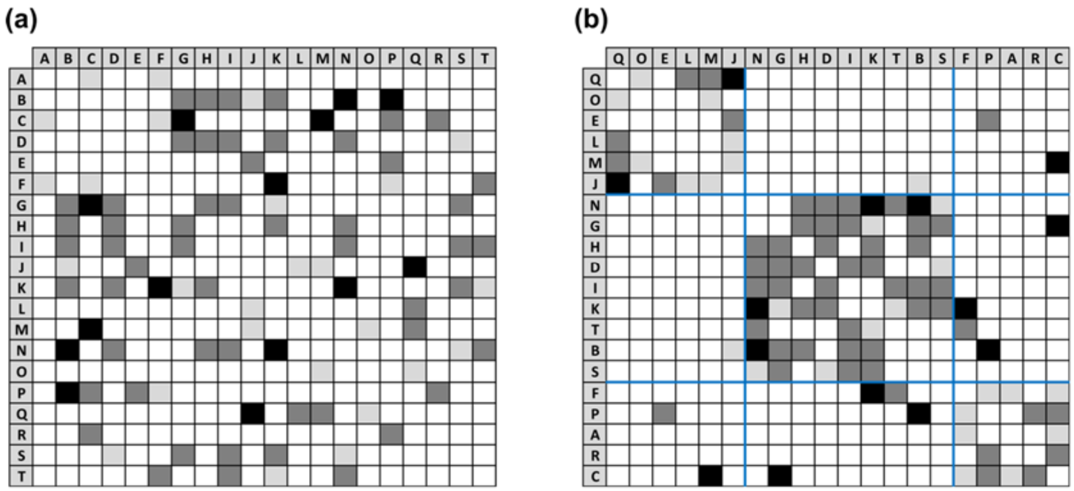


Figure 1. Hypothetical SV-3 in Its Original (Panel (a)) and Isomorphic (Panel (b)) Representations, Where Subsystems Have Been Permuted Into Architectural Communities
(Dabkowski et al., 2014)

More granularly, we need to answer three questions:

(Q1) How many subsystems should U connect to (degree, m)?;

(Q2) If U connects to m subsystems, which *m* subsystems should it connect to (adjacency)?; and

(Q3) If U connects to a specific set of *m* subsystems, what should the complexity of these interfaces be (weights)?

Under the scenario of evolutionary growth versus revolutionary change, we make the fundamental assumption that the current architecture foretells the future architecture. In other words, the existing patterns and characteristics of the subsystems' interfaces in Figure 1 provide us with useful evidence for predicting the pattern and characteristics of the

---

[4] In the parlance of network science, undirected interfaces are symmetric with respect to the SV-3's main diagonal. In other words, the interface from subsystem i to subsystem j implies the same interface from subsystem j to subsystem i. For directed interfaces, symmetry is not required, and the implication does not hold.

interfaces U will generate. As reported in our earlier Acquisition Research Symposium paper titled “The Budding SV3: Estimating the Cost of Architectural Growth Early in the Life Cycle” (Dabkowski & Valerdi, 2014), making this assumption allows us to address (Q1) through (Q3) as follows:[5]

> (A1) Degree: To model a “rich-by-birth” effect, view the degree of U (M_U) as a random variable with a probability mass function (PMF) equal to the observed degree distribution of the existing system (Dorogovtsev & Mendes, 2003);
>
> (A2) Adjacency: To incorporate a “rich-get-richer” effect, utilize the Barabási-Albert preferential attachment (PA) model from network science, where the probability subsystem i attaches to subsystem U is a linear function of its degree (d_i) or $p_i = d_i / \sum_{j=1}^{N} d_j$ (Barabási & Albert, 1999); and
>
> (A3) Weights: To mimic the observed complexity in the existing architecture, cast the complexity of the interface between U and subsystem i (w_{iU}) as a conditional random variable, where the PMF for w_{iU} equates to the observed interface complexity distribution of subsystem i.

Furthermore, when searching for patterns in an MDAP’s architecture, the manner in which systems engineers typically architect systems should be taken into account. For instance, in *The Art of Systems Architecting*, Maier and Rechtin (2000) note that the “most important aggregation and partitioning heuristics are to *minimize external coupling* and *maximize internal cohesion* [emphasis added].” Accordingly, looking for clusters or communities of subsystems where the density of intra- versus inter-community interfaces is high seems reasonable, and applying the Girvan-Newman community detection heuristic (Girvan & Newman, 2002) to the SV-3 in Panel (a) of Figure 1 identifies three *architectural communities*. As seen in Panel (b) of Figure 1, when the MDAP’s subsystems are permuted by their community membership, the system’s underlying macrostructure appears to abide Maier and Rechtin’s (2000) heuristics. Exploiting these architectural communities in (A1) to (A3) yields the following mechanism for estimating the cost of connecting subsystem U to the existing architecture (Dabkowski et al., 2014):

For a specified, suitably large number of iterations (e.g., 10,000)[6]…

Preprocessing

1. Initialize the system as the current system,
2. Use Girvan-Newman (2002) to identify architectural communities,
3. Randomly assign U to community *j*,

---

[5] See Dabkowski et al. (2013) for additional details.

[6] When estimating the population mean of a random variable X (μ_X) using Monte Carlo simulation, the minimum number of iterations required is a function of (a) the researcher’s desired accuracy for the estimate, which varies depending on the context, and (b) the population variance (σ_X^2), which is normally unknown. Accordingly, the researcher typically runs an initial set of iterations to generate unbiased estimates of μ_X and σ_X^2 from which the minimum number of iterations can be calculated (i.e., via Driels & Shin, 2004)

Intracommunity Growth

4. Generate a realization for $M_{\text{U,intra}}$ given U is assigned to community j (m_{intra}),
5. Connect U to m_{intra} subsystems inside community j using the PA model,
6. For each interface established in (5), assign complexity ($w_{i\text{U,intra}}$),

Intercommunity Growth

7. Generate a realization for $M_{\text{U,inter}}$ given U is assigned to community j (m_{inter}),
8. Connect U to m_{inter} communities using the PA model, and
9. For each interface established in (8), assign complexity ($w_{i\text{U,inter}}$),

Cost Estimation

10. Estimate the cost for the augmented system using COSYSMO (PM_{NS}*),
11. Calculate the additional cost of adding subsystem U ($PM_{NS}^* - PM_{NS}$), and
12. Store results and return to (3).

Generalizing Beyond Architectural Communities via Blockmodeling

While the above algorithm has intuitive appeal, the SV-3 in Figure 1 is hypothetical, and this raises the following questions: "Do (A1) through (A3) adequately model the growth of real-world SV-3s, and do SV-3s actually harbor architectural communities?" In a recent paper, we address these questions using 24 different SV-3s from a wide variety of MDAPs (Dabkowski & Valerdi, 2016). First, with respect to (A1) and (A2), formal hypothesis testing suggested that using the observed degree distribution generated far too many interfaces and blindly applying the PA model was ill-advised. In fact, the PMF for an incoming subsystem's number of interfaces $(P(M = m))$ and the strength of preferential attachment β interact, which led us to identify and utilize an optimal set of $\{P(M = m), \beta\}$ pairs for each SV-3. Moving on to (A3), none of the real-world SV-3s we examined were valued; thus, the validity of using the observed interface complexity distribution to estimate future interface complexity could not be assessed. Finally, as regards architectural communities, less than 50% of the SV-3s exhibited community structure worth exploiting, suggesting a non-community version of the algorithm was necessary. Simply put, significant adjustments to our earlier algorithm were necessary, and these are documented in Dabkowski and Valerdi (2016).

Notwithstanding these refinements, restricting our attention to architectural communities may ignore other, more compelling macrostructures within the architecture. For example, consider the hypothetical SV-3 in Panel (a) of Figure 2.

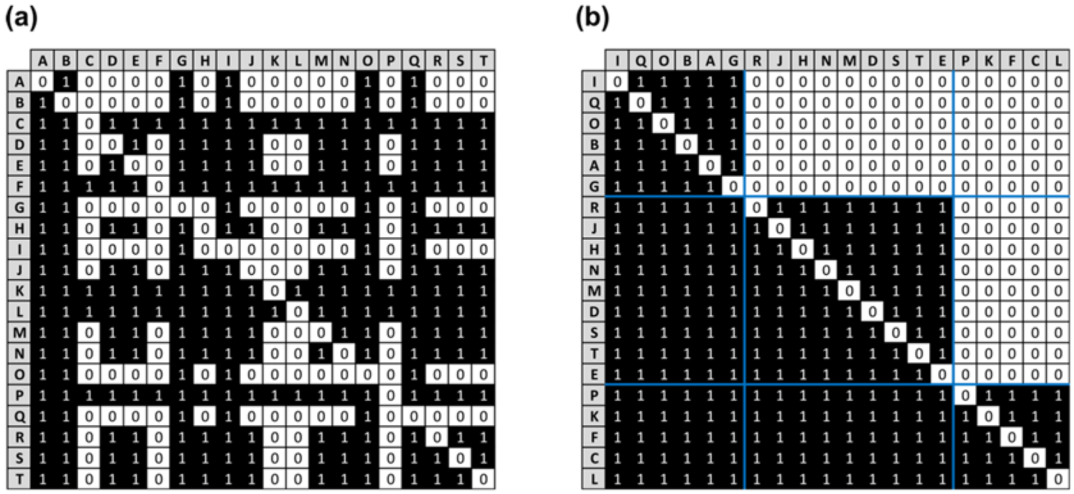


Figure 2. Hypothetical SV-3 With a Hierarchical Structure in Its Original (Panel (a)) and Isomorphic (Panel (b)) Representations, Where Subsystems Have Been Optimally Partitioned and Permuted

Consisting of $N = 20$ subsystems (labeled A through T) and $E = 251$ directed interfaces, the SV-3 is relatively dense, and, while the Girvan-Newman community detection heuristic identifies six architectural communities, the community structure is weak. Based on this result, we would invoke our non-community version of the algorithm. That said, the Girvan-Newman community detection heuristic was designed for sparse networks (Girvan & Newman, 2002), and the weak community structure may be spurious. Moreover, taking this approach would ignore the indisputable hierarchical structure of subsystems seen in Panel (b) of Figure 2, where subsystems in lower ranking clusters ({R, J, H, N, M, D, S, T, E} and {P, K, F, C, L}) not only have a high density of interfaces with subsystems inside their clusters but also have a high density of interfaces with subsystems inside higher ranking clusters.

To identify this and other hidden macrostructure, we can apply the network analysis technique known as blockmodeling, where a network consisting of $i = 1, \cdots, N$ objects (i.e., the SV-3 and its subsystems) is partitioned into $k = 1, \cdots, P$ nonoverlapping positions (or clusters) where the positions generally abide the structure represented in a $(P \times P)$ image matrix such that $P \ll N$ Conceived by computational sociologists at Harvard in the mid-1970s (e.g., White, Boorman, & Breiger, 1976; Boorman & White, 1976), blockmodeling methods have been an active area of research for over 40 years, and they have been integrated into popular network analysis software such as UCINET (Borgatti, Everett, & Freeman, 2002), R's *igraph* package (Csárdi & Nepusz, 2006), and Pajek (Mrvar & Batagelj, 2013).

Notable among these is Pajek's inclusion of Doreian, Batagelj, and Ferligoj's (2005) direct approach, which employs a simple object relocation routine that minimizes the number of inconsistencies between the permuted, partitioned $(N \times N)$ adjacency matrix (i.e., the SV-3) and a corresponding$(P \times P)$ image matrix. Invoked in Pajek via the commands `Network → Create Partition → Blockmodeling`, we ran Doreian et al.'s (2005) direct approach on the hypothetical SV-3 in Panel (a) of Figure 2, and this yielded the image matrix and reduced graph seen in Panels (a) and (b) of Figure 3, respectively. With zero inconsistences, the solution's partition matches Panel (b) of Figure 2, and it is the unique

global optimum. As Figure 3 clearly demonstrates, unlike Girvan and Newman's (2002) community detection heuristic, Doreian et al.'s (2005) direct approach recovered the hierarchical clustering of subsystems.

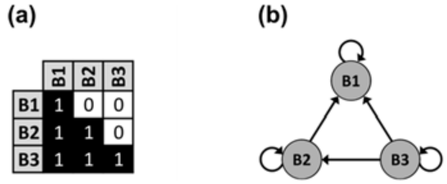


Figure 3. Globally Optimal Image Matrix (Panel (a)) and Reduced Graph (Panel (b)) for the Hypothetical SV-3 Seen in Panel (a) of Figure 2

In fact, blockmodeling can be seen as the natural generalization of community detection, as finding an optimal clustering of N objects into P communities is equivalent to finding the optimal partition of N objects for a P-position identity image matrix. For instance, consider the hypothetical SV-3 in Panels (a) and (b) of Figure 4.

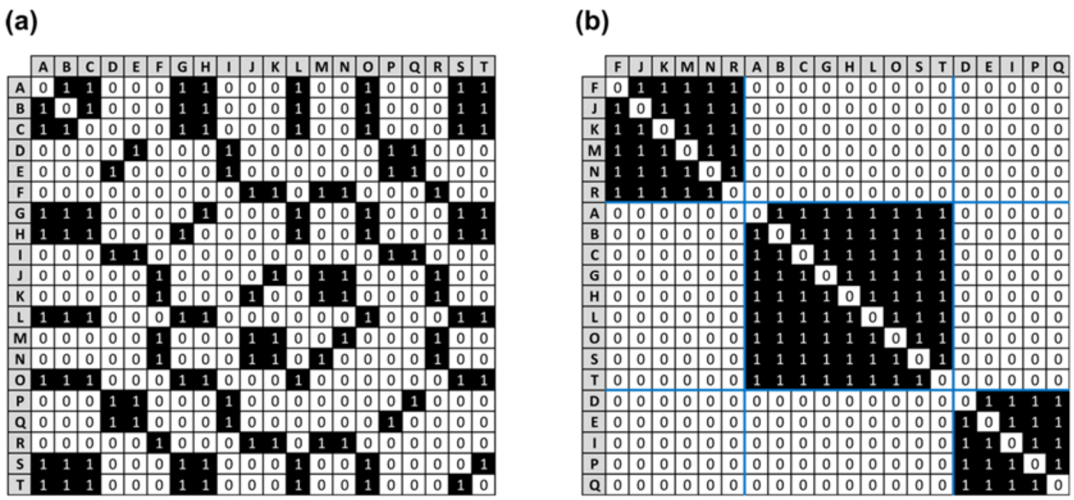


Figure 4. Hypothetical SV-3 With Community Structure in Its Original (Panel (a)) and Isomorphic (Panel (b)) Representations, Where Subsystems Have Been Optimally Partitioned and Permuted

With three isolated cliques and a sparse structure, we expect the Girvan-Newman community detection heuristic to identify the architectural communities, and it does. Similarly, Doreian et al.'s (2005) direct approach recovers the communities, yielding the globally optimal image matrix and reduced graph seen in Panels (a) and (b) of Figure 5, respectively.

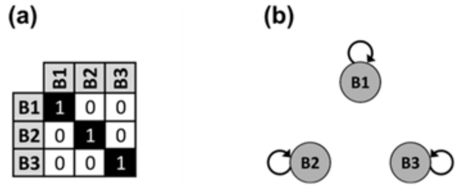


Figure 5. Globally Optimal Image Matrix (Panel (a)) and Reduced Graph (Panel (b)) for the Hypothetical SV-3 Seen in Panel (a) of Figure 4

Given these observations, the implication is that when it comes to identifying and exploiting the underlying macrostructure of a network, blockmodeling subsumes—and therefore trumps—community detection. Interestingly enough, however, this relationship has only recently been acknowledged by network scientists, as Newman and Leicht note in their 2007 paper extending earlier and more limited community detection methods:

> Here we describe a general technique for detecting structural features in large-scale network data that works by dividing the nodes of a network into classes such that the members of each class have similar patterns of connection to other nodes. … the idea is similar in philosophy to the block models proposed by White and others. (pp. 9564–9565)

Nonetheless, Doreian et al.'s (2005) direct approach is not a panacea, as it (1) generates locally optimal solutions and, thus, provides no guarantee that better fitting image matrices and partitions do not exist and (2) was designed to handle single one- or two-mode networks,[7] and, therefore, cannot readily accommodate multiple relations simultaneously. Unfortunately, both shortcomings are problematic. First, without a known optimality gap, we cannot definitively assess the quality of Pajek's solutions, and exact methods that generate global optima are necessary. Second, during our investigation of real-world SV-3s (Dabkowski & Valerdi, 2016), we discovered that 3 of the 24 SV-3s were actually mixed-mode networks. For example, consider the SV-3 in Figure 6, which consists of 10 internal subsystems and 7 external subsystems.

[7] One- and two-mode networks describe the connections that exist between a single set of objects and two distinct sets of objects, respectively. In the context of this paper, if an SV-3 is one-mode, the subsystems in its rows and columns are the same. If it is two-mode, they are different.

		Internal Subsystems										External Subsystems						
		I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	E1	E2	E3	E4	E5	E6	E7
Internal Subsystems	I1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	I2	1	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0
	I3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	I4	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	I5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	I6	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
	I7	0	1	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0
	I8	0	0	0	0	0	1	1	0	1	1	0	1	0	0	0	0	0
	I9	0	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1	1
	I10	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1

Figure 6. Multiple Relation Mixed-Mode SV-3 With 10 Internal Subsystems (Labeled I1 Through I10) and 7 External Subsystems (Labeled E1 Through E7)

In this SV-3, the 1-mode portion (located to the left of the vertical red line) shows the interfaces that exist between internal subsystems, where a 1 in cell (i, j) implies internal subsystem i interfaces with internal subsystem j. Similarly, the 2-mode portion (located to the right of the vertical red line) shows the interfaces that exist between internal and external subsystems, where a 1 in cell (i, m) implies internal subsystem i interfaces with external subsystem m. Clearly, each portion of the SV-3 contains valuable information for partitioning the internal subsystems, and we would like to include both in our analysis.

With this in mind, the first author embarked on a complementary line of research to develop an exact method for the blockmodeling of mixed-mode networks. Drawing on the integer programming approach of Brusco and Steinley (2009), this effort is chronicled in the "Exact Exploratory Blockmodeling of Multiple Relation, Mixed-Mode Networks Using Integer Programming" (Dabkowski, Fan, & Breiger, 2016), and it provides analysts with a reasonably efficient way to find globally optimal blockmodels for one-, two-, and mixed-mode SV-3s. Applying this method to the SV-3 in Figure 6 and capping the number of internal and external positons at three yields the results in Figure 7.

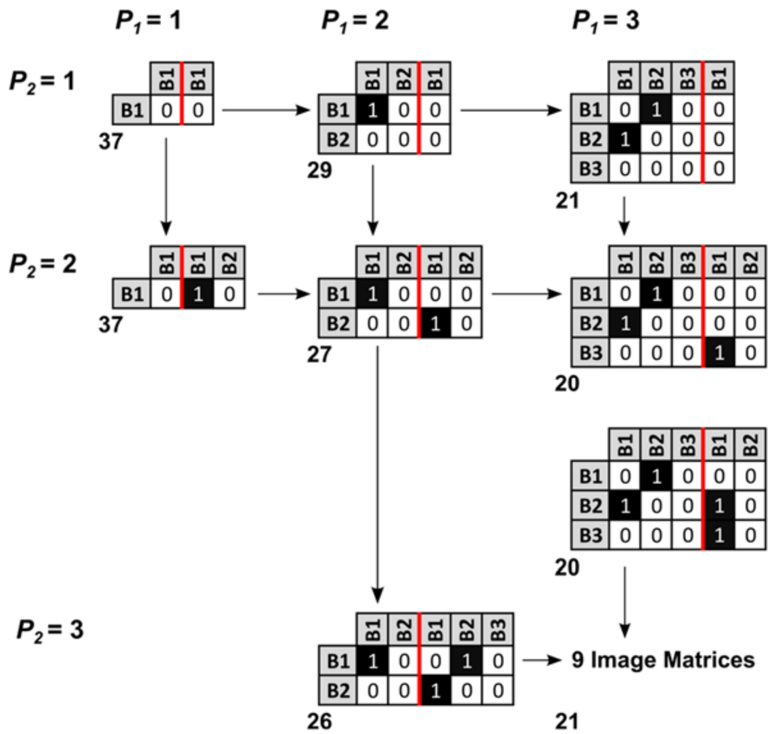


Figure 7. Globally Optimal Image Matrices for the Mixed-Mode SV-3 Seen in Figure 6, Where the Number of Inconsistencies Corresponding to the Globally Optimal $(P_1 \times P_1 | P_1 \times P_2)$ Image Matrix Is Given at the Bottom Left of the Matrix

As Figure 7 shows, with the exception of the $(3 \times 3 | 3 \times 3)$ image matrix, the minimum number of inconsistencies decreases monotonically as the number of internal or external positions increases, eventually reaching a minimum of 20 for the two globally optimal $(3 \times 3 | 3 \times 2)$ image matrices. Moreover, for each of the two globally optimal $(3 \times 3 | 3 \times 2)$ image matrices in Figure 7, the clustering of the internal and external subsystems is the same, and the corresponding permuted, partitioned network is given in Figure 8.

		Internal Subsystems										External Subsystems						
		I1	I6	I7	I9	I10	I2	I8	I3	I4	I5	E2	E1	E3	E4	E5	E6	E7
Internal Subsystems	I1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
	I6	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0
	I7	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	0
	I9	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	1	1
	I10	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	1	1
	I2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	I8	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0
	I3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	I4	1	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0
	I5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Figure 8. Mixed-Mode SV-3, Where the Rows and Columns Have Been Permuted According to the Globally Optimal $(3 \times 3 | 3 \times 2)$ Image Matrices and Partition in Figure 7

Interestingly, the clustering of internal subsystems appears to be entirely driven by connections outside the clusters. As with the hypothetical SV-3 in Figure 2, traditional

community detection algorithms cannot exploit this, and, as expected, Girvan and Newman's (2002) heuristic returned an insignificant, much different result using the one-mode portion of Figure 6. Nonetheless, as the number of positions increases the exact approach quickly becomes impractical, and mixed-mode blockmodeling heuristics are necessary. Accordingly, the first author built one in Pajek leveraging Doreian et al.'s (2005) direct approach, and its performance was outstanding, as it found the globally optimal solutions in a reasonable amount of time.

Integrating Results

Equipped with exact and heuristic methods for the blockmodeling of SV-3s, we can replace Step (2) in our earlier algorithm ("Use Girvan-Newman (2002) to identify architectural communities") with "Use Dabkowski-Fan-Breiger (2015; 2016) to identify an optimal P-position image matrix and partition of subsystems." If the optimal image matrix and partition suggest a compelling architectural structure, future evolutionary growth should abide it, and, similar to our earlier algorithm, we can randomly assign an incoming subsystem (X) to position k. However, unlike our earlier algorithm, the assignment of subsystem X's m interfaces to positions is no longer modeled via separate PMFs for each position (or community). It is the sum of m independent and identically distributed categorical random variables, where the probability interface j for $j = 1, \cdots, m$ links to a subsystem in position l for $l = 1, \cdots, P$ is given by:

$$\frac{\text{number of interfaces in block } (k,l) \text{ of the partitioned and permuted SV-3}}{\text{number of interfaces in row } k \text{ of the partitioned and permuted SV-3}} \tag{1}$$ [8]

As such, the collective assignment of subsystem X's m interfaces to positions can be modeled as a random $(1 \times P)$ vector $\boldsymbol{C}$, where $\boldsymbol{C}$ follows a Multinomial $m, \boldsymbol{p}$ distribution and $\boldsymbol{p}$ is the $(1 \times P)$ vector of multinomial probabilities defined in (Equation 1).

Of course, $\boldsymbol{C}$ could generate a realization ($\boldsymbol{c}$) where one or more of its elements ($\boldsymbol{c}_l$) exceeds the number of subsystems in its respective position (N_l). In this case, we can apply the following numerical recipe to generate a feasible realization for $\boldsymbol{C}$: (1) for all positions where $\boldsymbol{c}_l \geq N_l$,, aggregate the $\boldsymbol{c}_l - N_l$ excess interfaces into an accumulator variable, m', and set $\boldsymbol{c}_l$ as N_l; (2) remove these positions and their probability mass from $\boldsymbol{C}$; (3) renormalize the multinomial probabilities; and (4) redistribute the m' excess interfaces among the remaining positions, iterating as necessary.

Integrating these adjustments, as well as refinements from Dabkowski and Valerdi (2016), into our earlier algorithm yields the modified pseudocode below:

For a specified, suitably large number of iterations …

[8] As Kolaczyk and Csárdi (2014) note, in a nonstochastic blockmodel, "the edge probabilities π_{qr} [where q and r represent positions], and the maximum likelihood estimates—which are natural here—are simply the corresponding empirical frequencies" (p. 97).

Preprocessing

1. Initialize the system as the current system,
2. Build an optimal set of $\{P(M = m), \beta\}$ pairs,
3. Use Dabkowski-Fan-Breiger (2015; 2016) to identify an optimal P-position image matrix and partition of subsystems,

Growth

4. Randomly select a member from the optimal set of $\{P(M = m), \beta\}$ pairs,
5. Generate a realization for the incoming subsystem's (X's) number of interfaces using $P(M = m)$; if the optimal image matrix and partition suggest a compelling architectural structure, use *Connection Option A*; otherwise, use *Connection Option B*,

Connection Option A

6a. Randomly assign X to position k,

6b. Model the collective assignment of subsystem X's m interfaces to positions as a random $(1 \times P)$ vector $\boldsymbol{C}$, where $\boldsymbol{C}$ follows a Multinomial $(m, \boldsymbol{p})$ distribution and $\boldsymbol{p}$ is the $(1 \times P)$ vector of multinomial probabilities given by (Equation 1); generate a feasible realization for $\boldsymbol{C}$,

6c. For $l = 1, \cdots, P$, attach X to $\boldsymbol{c}_l$ subsystems inside position l using attachment probabilities $p_i = {d_i}^{\beta} / \sum_{j=1}^{N} {d_j}^{\beta}$,

6d. For each interface established in (6c), assign complexity (w_{iX}),

Connection Option B

6a. Attach X to m subsystems using attachment probabilities $p_i = {d_i}^{\beta} / \sum_{j=1}^{N} {d_j}^{\beta}$,

6b. For each interface established in (6a), assign complexity (w_{iX}),

Cost Estimation

7. Estimate the cost for the augmented system using COSYSMO (PM_{NS}*),
8. Calculate the additional cost of adding subsystem X (PM_{NS}* − PM_{NS}), and
9. Store results and return to (4).

As seen above, unlike our previous algorithm, *Connection Option B* provides an alternative, nonposition-based growth mechanism. Additionally, *Connection Option A* does not condition interface complexities based on the connected subsystems' positions of assignment (i.e., $w_{iX,l}$), as any patterns in intra- or interposition complexity could be due to chance. Specifically, the blockmodeling methods developed in Dabkowski et al. (2015; 2016) are for unvalued networks. Therefore, the statistical significance of apparent structure in the interface complexities must be assessed prior to leveraging them in the algorithm.

Using our improved algorithm, we can estimate the cost of unforeseen, internal architectural growth in mixed-mode SV-3s (as well as one- and two-mode SV-3s). For example, assume the system represented in Figure 6 has the following values for COSYSMO's parameters: $A = 0.25$; $E = 1.06$; $\prod_{j=1}^{14} EM_j = 0.89$; and 75 easy, 50 nominal, and 10 difficult requirements. Additionally, if we assume its interface complexities are portrayed in Figure 9, the system has 12 interfaces between internal subsystems (6 easy, 5 nominal, and 1 difficult) and 13 interfaces between external subsystems (6 easy, 6 nominal, and 1 difficult). Using COSYSMO's CER and weights from Valerdi (2008), we estimate that 59.24 PM_{NS} of systems engineering effort are required to successfully conceptualize,

develop, and test the MDAP. At this point, we have initialized the system as the current system, and Step (1) of the algorithm is complete.

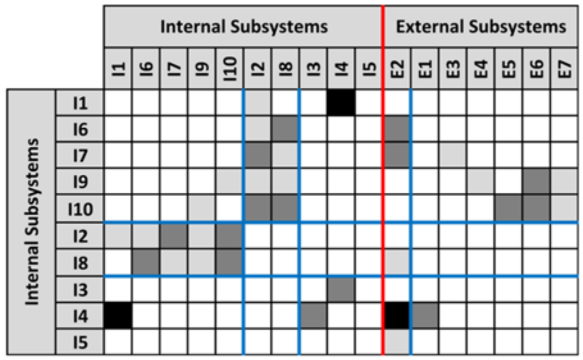
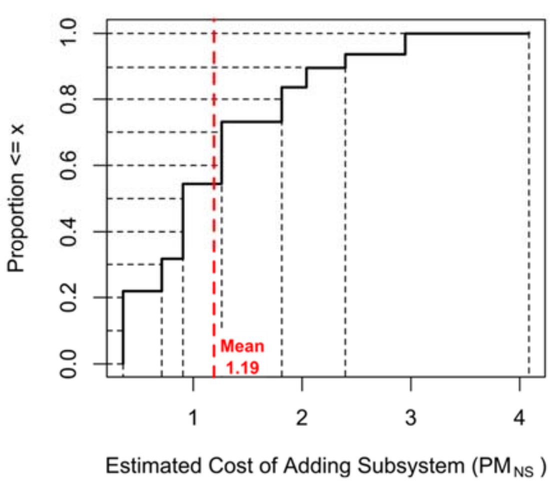


Figure 9. Hypothetical Interface Complexities for the System Represented in Figure 6, Where Cell (i, j) Is Shaded if Subsystem i Interfaces With Subsystem j, and Darker Shades Indicate Greater Interface Complexity (i.e., Light Gray ⇒ Easy, Medium Gray ⇒ Nominal, Black ⇒ Difficult)

Our next task is to build an optimal set of $\{P(M = m), \beta\}$ pairs. Using our approach in Dabkowski and Valerdi (2016), there are five feasible PMFs for m. Among these, the single optimum is $P(M = 2) = 0.5$ and $P(M = 1) = 0.5$, and the corresponding optimal set of β is {0, …, 0.4}.

With Step (2) complete, our last preprocessing step is to identify an optimal P-position image matrix and partition of subsystems, and the global optimal solution is given in Figure 9. This result, along with the optimal set of $\{P(M = m), \beta\}$ pairs, is then ingested into a Monte Carlo simulation, which performs Steps (4) through (9). Running the simulation for 10,000 iterations yields the results seen in Figure 10.



Percentile	PM_{NS}	Percentile	PM_{NS}
5%	0.36	55%	1.26
10%	0.36	60%	1.26
15%	0.36	65%	1.26
20%	0.36	70%	1.26
25%	0.71	75%	1.81
30%	0.71	80%	1.81
35%	0.91	85%	2.04
40%	0.91	90%	2.40
45%	0.91	95%	2.95
50%	0.91	100%	4.08

Figure 10. Empirical Cumulative Distribution Function and Percentiles for the Estimated Cost of Connecting an Additional Subsystem to the Internal Subsystems of Figure 9

As seen in Figure 10, the expected cost to connect an additional subsystem (X) to the internal subsystems of Figure 9 is 1.19 PM_{NS}, and the associated 95% confidence

interval is (1.177, 1.206) PM_{NS}. Moreover, although the maximum cost to attach subsystem X should not exceed 4.08 PM_{NS}, there is only a 5% chance it will be more than 2.95 PM_{NS}. Finally, if we condition our estimate on X's position of assignment, the expected cost in person months (nominal schedule) is 1.00, 1.05, and 1.53 for positions 1, 2, and 3, respectively. In the absence of additional information, these estimates represent our "best guess" for the cost to attach a new subsystem to the existing architecture, and they help to quantify the likelihood of excessive cost growth.

Limitations and Future Work

Although our use of blockmodeling to identify and exploit an SV-3's globally optimal macrostructure provides a useful generalization, the algorithm and its supporting methods have several limitations, and these represent opportunities for future research. Starting with insufficient data, SV-3s are not currently weighted by interface complexity, and the validity of using the observed interface complexity distribution to estimate future interface complexity could not be assessed. Accordingly, sponsored research is required to generate the necessary data for statistical investigation.

Moving on to the algorithm's internal steps, Connection Option A assigns incoming subsystems to positions using a uniform distribution. If we assume unforeseen architectural growth is equally likely in all positions, this is appropriate. That said, other possibilities are worth exploring. For example, the probability subsystem X is assigned to position k could be modeled as either a function of position k's size or a function of subsystem X's number of interfaces. Additionally, although the algorithm is currently limited to estimating internal architectural growth, modifying it to address external architectural growth is natural, especially when we consider that its optimal macrostructure was obtained from the interfaces between its internal and external subsystems.

Finally, in a more general sense, mixed-mode blockmodeling remains a fruitful area for future research, as it suffers from scalability challenges, especially as the number of internal and external positions grow. Possible solutions to address this include improved integer programming formulations and the use of high throughput/high performance computing.

Conclusion

MDAPs are notoriously prone to cost overruns and schedule delays, and requirements volatility is at least partially to blame. In particular, when the desired capabilities of a system change during its life cycle, substantial reengineering and cost growth can result, especially when a new subsystem must be incorporated into an existing architecture. Of course, the likelihood and specifics of such additions are rarely known ahead of time, and predicting integration costs is challenging.

In this paper, we presented a novel algorithm to address this issue. Specifically, leveraging an integer programming implementation of the social network analysis technique blockmodeling, we optimally partitioned the subsystems represented in the SV-3 into architectural positions. Using this abstracted structure, we subsequently grew the architecture according to its statistical properties, and we estimated this unforeseen cost of evolutionary architectural growth via COSYSMO. Although our approach has limitations, the algorithm provides a useful prototype for pre–MS A cost risk analysis, and it continues to reinforce the potential of viewing DoDAF's models as computational objects.

References

Barabási, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, *286*(5439), 509–512. doi:10.1126/science.286.5439.509

Blanchard, B., & Fabrycky, W. (1998). *Systems engineering and analysis* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Bolten, J. G., Leonard, R. S., Arena, M. V., Younossi, O., & Sollinger, J. M. (2008). Sources of weapon system cost growth: Analysis of 35 major defense acquisition programs. Santa Monica, CA: RAND. Retrieved from http://www.rand.org/content/dam/rand/pubs/monographs/2008/RAND_MG670.pdf

Boorman, S. A., & White, H. C. (1976). Social structure from multiple networks. II. Role structures. *American Journal of Sociology, 81*(6), 1384–1446. Retrieved from http://www.jstor.org/stable/2777009

Borgatti, S. P., Everett, M. G., & Freeman, L. C. (2002). Ucinet for Windows: Software for social network analysis (Ver. 6.488) [Software]. Harvard, MA: Analytic Technologies. Retrieved from http://www.analytictech.com/ucinet

Brusco, M., & Steinley, D. (2009). Integer programs for one- and two-mode blockmodeling based on prespecified image matrices for structural and regular equivalence. *Journal of Mathematical Psychology, 53*(6), 577–585. doi:10.1016/j.jmp.2009.08.003

CJCS. (2012, January 19). *Manual for the operation of the Joint Capabilities Integration and Development System.* Retrieved from https://dap.dau.mil/policy/Documents/2012/JCIDS%20Manual%2019%20Jan%202012.pdf

Csárdi, G., & Nepusz, T. (2006). *The igraph software package for complex network research* (Ver. 1.0.0) [Software]. Retrieved from http://igraph.org

Cole, R. (2012, October). *Proxy estimation costing for systems (PECS).* Paper presented at the 27th International Forum on COCOMO and Systems/Software Cost Modeling, Software Engineering Institute, Pittsburgh, PA. Retrieved from http://csse.usc.edu/csse/event/2012/COCOMO/presentations/CIIforum_2012_Reggie_Cole.pptx

Dabkowski, M., Fan, N., & Breiger, R. (2015). *Exploratory blockmodeling for one-mode, unsigned, deterministic networks using integer programming*. Manuscript submitted for publication.

Dabkowski, M., Fan, N., & Breiger, R. (2016). *Exact exploratory blockmodeling of multiple relation, mixed-mode networks using integer programming*. Manuscript in preparation.

Dabkowski, M., Reidy, B., Estrada, J., & Valerdi, R. (2013). Network science enabled cost estimation in support of model-based systems engineering. *Procedia Computer Science, 16*, 89–97. doi:10.1016/j.procs.2013.01.010

Dabkowski, M., & Valerdi, R. (2014, April). The budding SV3: Estimating the cost of architectural growth early in the life cycle. In D. A. Nussbaum (Chair of Panel 22—Enhancing Cost Estimating Techniques), *Proceedings of the 11th Annual Acquisition Research Symposium*. Retrieved from http://www.acquisitionresearch.net/files/FY2014/NPS-AM-14-C11P22R02-076.pdf

Dabkowski, M., & Valerdi, R. (2016). *Modeling evolutionary architectural growth in major defense acquisition programs.* Manuscript submitted for publication.

Dabkowski, M., Valerdi, R., & Farr, J. (2014). Exploiting architectural communities in early life cycle cost estimation. *Procedia Computer Science, 28*, 95–102. doi:10.1016/j.procs.2014.03.013

Doreian, P., Batagelj, V., & Ferligoj, A. (2005). *Generalized blockmodeling*. New York, NY: Cambridge University Press.

Dorogovtsev, S. N., & Mendes, J. F. F. (2003). Non-equilibrium networks. In *Evolution of networks: From biological nets to the Internet and WWW*. New York, NY: Oxford University Press. doi:10.1093/acprof:oso/9780198515906.003.0006

Driels, M. R., & Shin, Y. S. (2004, April). *Determining the number of iterations for Monte Carlo simulations of weapon effectiveness* (Technical Report No. NPS-MAE-04-005). Retrieved from http://www.dtic.mil/dtic/tr/fulltext/u2/a423541.pdf

Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences, 99*(12), 7821–7826. doi:10.1073/pnas.122653799

GAO. (2011). *DoD cost overruns: Trends in Nunn-McCurdy breaches and tools to manage weapon systems acquisition costs*. Retrieved from http://www.gao.gov/assets/130/125861.pdf

GAO. (2015). *Defense acquisitions: Assessments of selected weapon programs.* Retrieved from http://www.gao.gov/assets/670/668986.pdf

Honour, E. C. (2004). *Understanding the value of systems engineering.* Retrieved from http://www.seintelligence.fr/content/images/2015/12/ValueSE-INCOSE04.pdf

Kolaczyk, E. D., & Csárdi, G. (2014). *Statistical analysis of network data with R.* New York, NY: Springer.

Maier, M., & Rechtin, E. (2000). *The art of systems architecting* (2nd ed.). New York, NY: CRC Press.

Mrvar, A., & Batagelj, V. (2013). Pajek64: Program for analysis and visualization of very large networks (Ver. 3.12) [Software]. Retrieved from http://mrvar.fdv.uni-lj.si/pajek

Newman, M. E., & Leicht, E. A. (2007). Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences, 104*(23), 9564–9569. doi:10.1073/pnas.0610537104

Peña, M., & Valerdi, R. (2015). Characterizing the impact of requirements volatility on systems engineering effort. *Systems Engineering, 18*(1), 59–70. doi:10.1111/sys.21288

Under Secretary of Defense for Acquisition, Technology, and Logistics (USD[AT&L]). (2013, November 25). *Operation of the defense acquisition system* (DoD Instruction 5000.02). Washington, DC: Author.

Valerdi, R. (2008). *The Constructive Systems Engineering Cost Model (COSYSMO): Quantifying the costs of systems engineering effort in complex systems.* Saarbrücken, Germany: VDM Verlag.

Valerdi, R., Dabkowski, M., & Dixit, I. (2015). Reliability improvement of major defense acquisition program cost estimates—Mapping DoDAF to COSYSMO. *Systems Engineering, 18*(5), 530–547. doi:10.1002/sys.21327

Wang, G., Valerdi, R., Roedler, G., Ankrum, A., & Gaffney, J. (2012). Harmonising software engineering and systems engineering cost estimation. International Journal of Computer *Integrated Manufacturing, 25*(4–5), 432–443. doi:10.1080/0951192X.2010.542182

Weapon Systems Acquisition Reform Act of 2009 (WASRA), Pub. L. No. 111-23. 123 Stat. 1704 (2009).

White, H. C., Boorman, S. A., & Breiger, R. L. (1976). Social structure from multiple networks. I. Blockmodels of roles and positions. *American Journal of Sociology, 81*(4), 730–780. Retrieved from http://www.jstor.org/stable/2777596

Acknowledgments

This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-13-1-0032, and the Office of the Secretary of Defense.

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CA 93943

www.acquisitionresearch.net

Blockmodeling and the Estimation of Evolutionary Architectural Growth in Major Defense Acquisition Programs

*LTC Matthew Dabkowski**
Dr. Ricardo Valerdi

Department of Systems & Industrial Engineering
College of Engineering
University of Arizona

* mfd1@email.arizona.edu

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-13-1-0032, and the Office of the Secretary of Defense. The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Agenda

- Purpose, Question, and Contribution
- Challenge and Opportunity in Pre-MS A Cost Analysis
- Mapping DoDAF to COSYSMO
- Leveraging SE and Exploiting the SV-3
- Estimating Unforeseen Architectural Growth in MDAPs
  – Microstructure
  – Macrostructure
- Simulating Growth and Estimating Cost
- “Blockmodeling” Beyond Architectural Communities
- Future work
- Questions

Overarching Purpose: To transform Model-Based System Engineering (MBSE) artifacts into computational knowledge that can be leveraged early in the system lifecycle when uncertainty is high and confidence is low

Focused Question: Can parametric cost estimation, in conjunction with DoD Architecture Framework (DoDAF) models, capture the monetary impact of architectural changes early in the system lifecycle?

Principal Contribution: A network science-based algorithm for estimating the cost of unforeseen architectural growth

Challenge and Opportunity in Pre-MS A Cost Analysis

We find ourselves in challenging times . . .

- Sequestration in 2013 + CRs =
  Reduced production +
  Hard modernization decisions +
  ⋯ + **Difficult cost planning**

. . . and times were already tough . . .

- 1997-2009: 47 MDAPs had cost overruns of at least 15%/30% over their current/baseline estimates

. . . especially early in the life cycle . . .

- ~ 28% of a system's baseline requirements will change

. . . as late adds carry substantial costs.

- 2014: 6 of 14 largest cost overruns due to new capabilities

But there is an appetite for change . . .

- *WSARA* (2009): Increased the rigor of Pre-MS A cost analysis (baseline shifted from MS B to MS A)
- *DoDI 5000.02* (2013): Mandated a draft CDD, with required DoDAF models, be submitted Pre-MS A

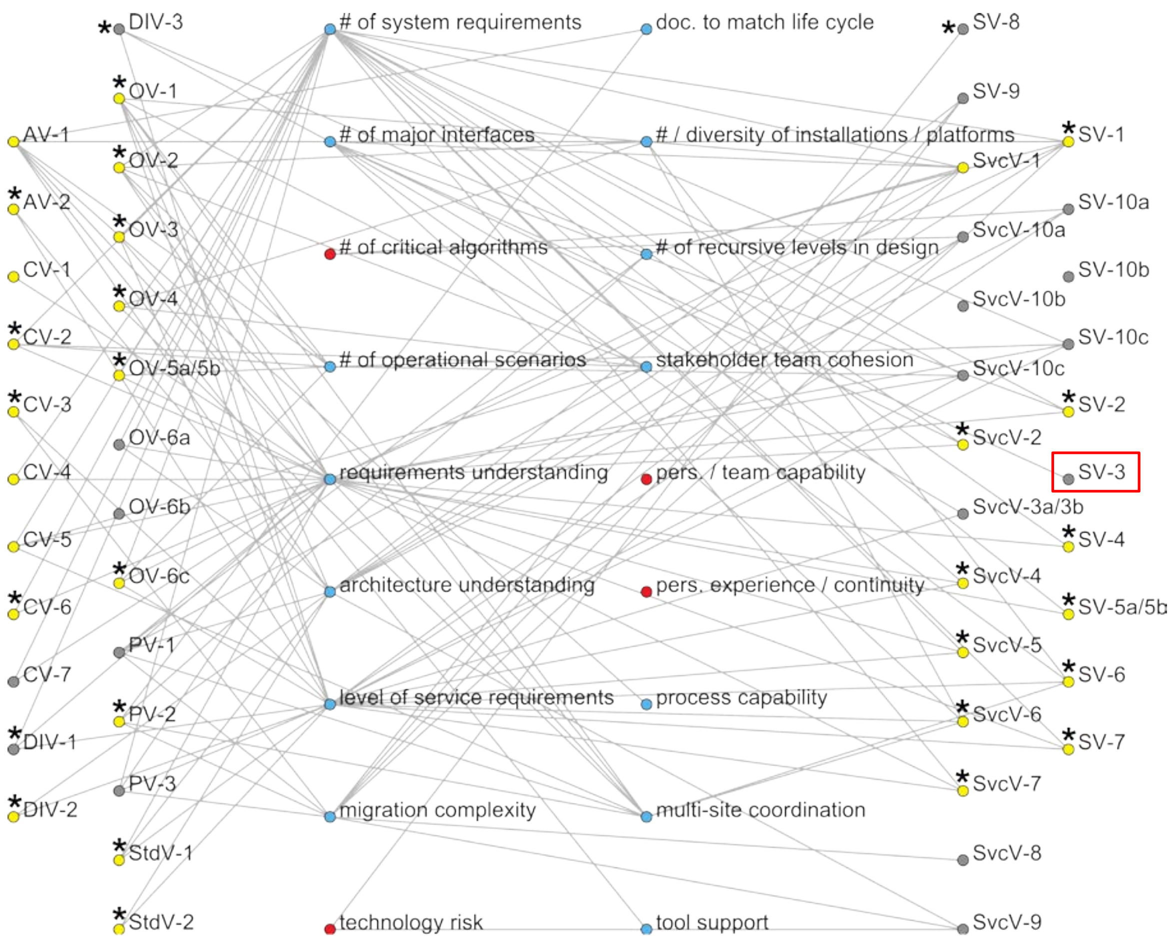
. . . and this presents an opportunity.

- DoDAF includes factors that influence system engineering (SE) effort (e.g., interfaces)
- COSYSMO estimates SE effort

DoDAF's models appear to map to COSYSMO's parameters

azengineering

DoDAF models required Pre-MS A nearly span COSYSMO's drivers*



78% of COSYSMO's drivers map to DoDAF models submitted early in the life cycle

Legend

- DoDAF model X is relevant for rating COSYSMO driver Y
- Model required Pre-MS A (2012-2015)
* - Model required Pre-MS A (2015-)
- Pre-MS A model(s) maps to driver
- No Pre-MS A model maps to driver

AV - All (2 models)
CV - Capability (7 models)
DIV - Data and information (3 models)
OV - Operational (9 models)
PV - Project (3 models)
StdV - Standards (2 models)
SvcV - Services (13 models)
SV - Systems (13 models)

* Valerdi, R., Dabkowski, M., & Dixit, I. (2015). Reliability Improvement of Major Defense Acquisition Program Cost Estimates – Mapping DoDAF to COSYSMO. *Systems Engineering*, 18(5), 530-547. doi:10.1002/sys.21327

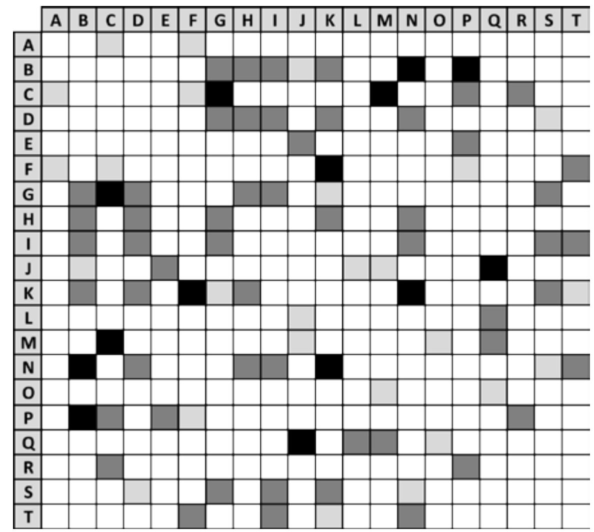
Leveraging SE and Exploiting the SV-3

- From the 2008 National Research Council report "Pre-Milestone A and Early-Phase Systems Engineering" . . .
  - The "application of SE to decisions made in the pre-Milestone A period is critical to avoiding (or at least minimizing) cost and schedule overruns" (p. 3)
  - 3 of the 6 primary drivers of cost growth addressable by SE are:
    1. Incomplete requirements at MS B,
    2. System complexity (via internal, architectural design), and
    3. External interface complexity (via network-centric operations or "systems of systems" constructs) (pp. 82-85)
- The SV-3 (or Systems-Systems Matrix) provides an abstraction of all 3, as requirements (however incomplete) drive the selection of subsystems (nodes) which are connected by interfaces (edges), both internal and external

Formally evaluating the SV-3 Pre-MS A and estimating its potential growth holds promise for minimizing cost overruns

Hypothetical SV-3

- 20 subsystems with 47 interfaces of varying complexity
- Without loss of generality, assume there are . . .
  - 200 easy, 200 nominal, and 50 difficult requirements
  - 5 difficult critical algorithms
- Using additional w_{ik} and EM_j data,* apply CER to obtain an initial estimate of PM_{NS}



SV-3

Interface Complexity

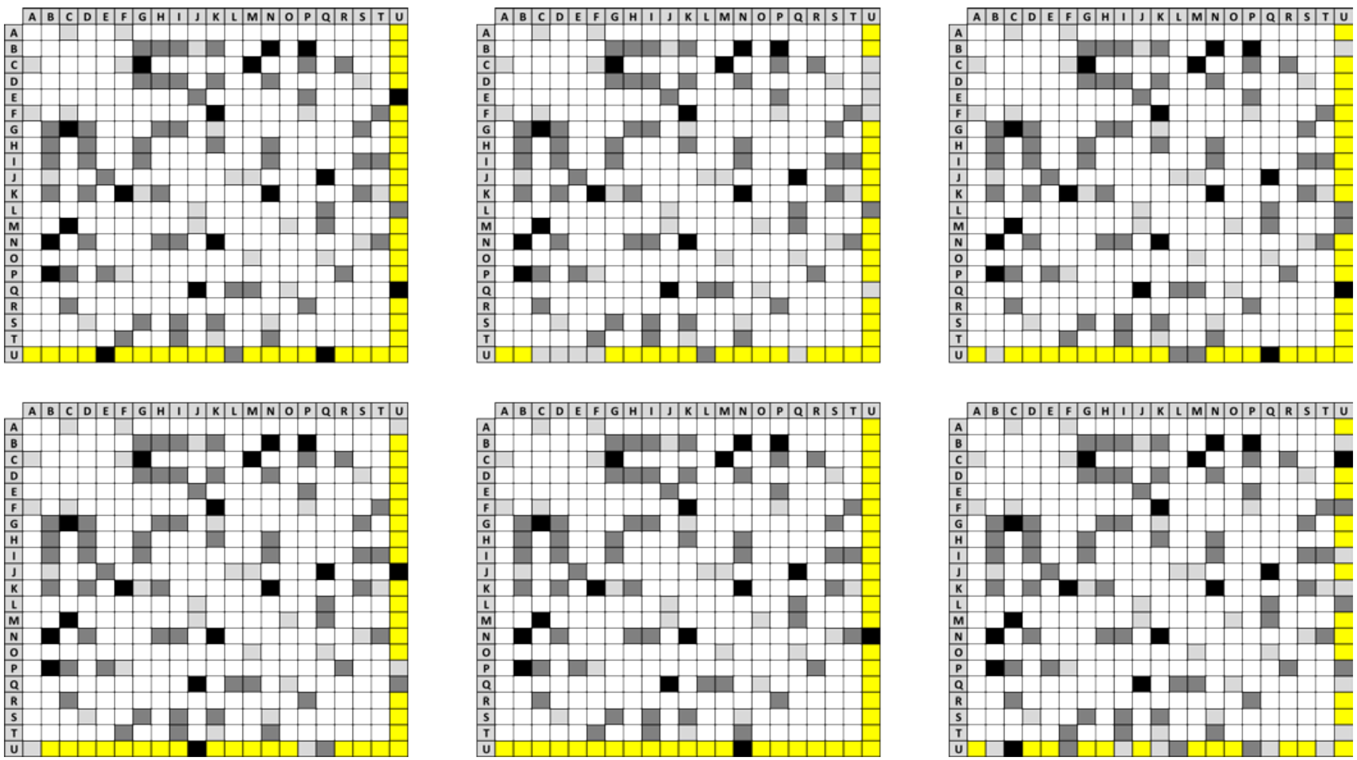
☐ = Easy, ☐= Nominal, ■ = Difficult

$$PM_{NS} = 0.25 \cdot \left(\underbrace{(0.5 \times 200 + 1.0 \times 200 + 5.0 \times 50)}_{\text{requirements}} + \underbrace{(11.5 \times 5)}_{\text{algorithms}} + \underbrace{(1.1 \times 13 + 2.8 \times 27 + 6.3 \times 7)}_{\text{interfaces}} \right)^{1.06} \cdot 0.89 = 245.27$$

* Valerdi, R. (2008). *The Constructive Systems Engineering Cost Model (COSYSMO): Quantifying the Costs of Systems Engineering Effort in Complex Systems*. Saarbrücken, Germany: VDM Verlag.

What about inevitable, unforeseen change?

- This is Pre-MS A ⇒ <u>requirements will change</u>



If we add a new subsystem U to the existing architecture, how will it connect?

What will it cost?

The analytical task

Estimate the number of interfaces (by complexity level) U will generate

(Q1) How many subsystems should **U** connect to (**degree, *m***)?,

(Q2) If **U** connects to *m* subsystems, which *m* subsystems should it connect to (**adjacency**)?, and

(Q3) If **U** connects to a specific set of *m* subsystems, what should the complexity of these interfaces be (**weights**)?

Network Science – A mechanism for generating unforeseen architectural growth (microstructure)

Fundamental assumption: Current architecture foretells future architecture

Degree: Treat degree of **U** (M_U) as a random variable with PMF equal to the degree distribution of the existing system ("rich-by-birth") (Dorogovtsev & Mendes, 2003)

m	2	4	5	6	7	8
n_m	5	3	5	3	3	1
$P(M_U = m)$	0.25	0.15	0.25	0.15	0.15	0.05

Adjacency: Utilize Barabási–Albert (1999) preferential attachment (PA) model, where highly connected subsystems are more likely to interface with **U** ("rich-get-richer")

System (i)	A	B	C	D	E	F	G	H	I	J
d_i	2	7	6	6	2	5	7	5	6	5
p_i	0.021	0.074	0.064	0.064	0.021	0.053	0.074	0.053	0.064	0.053
System (i)	K	L	M	N	O	P	Q	R	S	T
d_i	8	2	4	7	2	5	4	2	5	4
p_i	0.085	0.021	0.043	0.074	0.021	0.053	0.043	0.021	0.053	0.043

$$p_i = \frac{d_i}{\sum_{j=1}^{N} d_j}$$

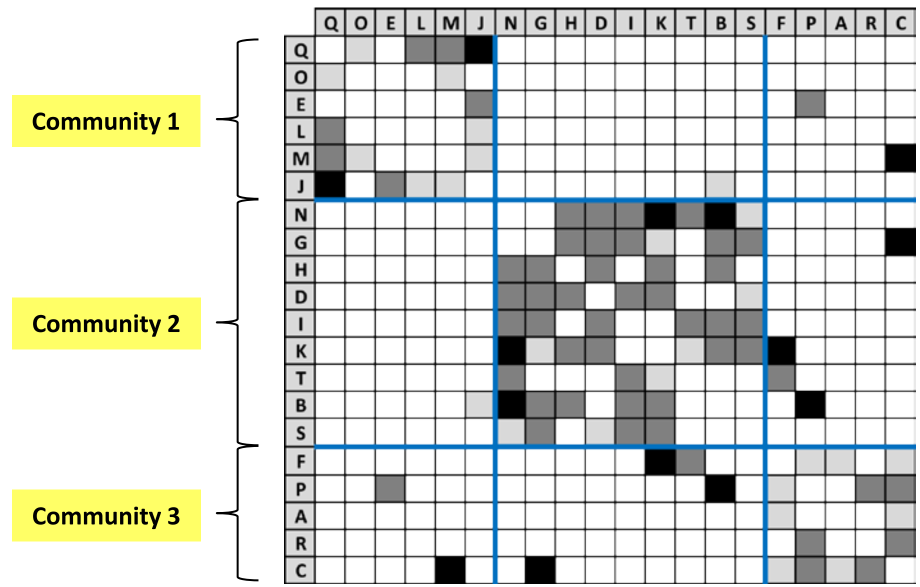
Weights: Model complexity of the interface between **U** and subsystem ***i*** (w_{iU}) as a random variable, where the pmf for w_{iU} is ***i***'s interface complexity distribution

Network Science – A mechanism for generating unforeseen architectural growth (macrostructure)

Fundamental assumption: Current architecture foretells future architecture

From *The Art of Systems Architecting*: "The most important aggregation and partitioning heuristics are to **minimize external coupling** and **maximize internal cohesion**"*

Architectural communities: Utilize Girvan-Newman (2002) to identify groups of subsystems such that the number of interfaces is sparse between and dense within groups



Intracommunity Community	Δ	Intercommunity Communities	Δ
1: {Q, O, E, L, M, J}	0.5333	1 and 2	0.0095
2: {N, G, H, D, I, K, T, B, S}	0.6944	1 and 3	0.0364
3: {F, P, A, R, C}	0.7000	2 and 3	0.0440

* Maier, M., & Rechtin, E. (2000). *The Art of Systems Architecting*. (2nd ed.). New York, NY: CRC Press.

Simulating Growth and Estimating Cost – Dabkowski et al. (2014)

For a specified number of iterations . . .

Preprocessing

1. Initialize the system as the current system
2. Use Girvan-Newman (2002) to identify architectural communities
3. Randomly assign **U** to community k

Intracommunity Growth

4. Generate a realization for $M_{\mathbf{U},intra}$ given **U** is assigned to community k (m_{intra})
5. Connect **U** to m_{intra} subsystems inside community k using the BA model
6. For each interface established in (5), assign complexity ($w_{i\mathbf{U},intra}$)

Intercommunity Growth

7. Generate a realization for $M_{\mathbf{U},inter}$ given **U** is assigned to community k (m_{inter})
8. Connect **U** to m_{inter} communities using the BA model
9. For each interface established in (8), assign complexity ($w_{i\mathbf{U},inter}$)

Cost Estimation

10. Estimate cost for augmented system using COSYSMO (PM_{NS}*)
11. Calculate additional cost of adding subsystem **U** ($PM_{NS}^* - PM_{NS}$)
12. Store results and return to (3)

"Did I build the <u>right</u> model? ***Is it general enough?"***

Community detection may miss key macrostructure . . .

Hypothetical SV-3

Original representation

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	0	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
B	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
C	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
D	1	1	0	0	1	0	1	1	1	1	0	0	1	1	1	0	1	1	1	1
E	1	1	0	1	0	0	1	1	1	1	0	0	1	1	1	0	1	1	1	1
F	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0
H	1	1	0	1	1	0	1	0	1	1	0	0	1	1	1	0	1	1	1	1
I	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0
J	1	1	0	1	1	0	1	1	1	0	0	0	1	1	1	0	1	1	1	1
K	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
L	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
M	1	1	0	1	1	0	1	1	1	1	0	0	0	1	1	0	1	1	1	1
N	1	1	0	1	1	0	1	1	1	1	0	0	1	0	1	0	1	1	1	1
O	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0
P	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
Q	1	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
R	1	1	0	1	1	0	1	1	1	1	0	0	1	1	1	0	1	0	1	1
S	1	1	0	1	1	0	1	1	1	1	0	0	1	1	1	0	1	1	0	1
T	1	1	0	1	1	0	1	1	1	1	0	0	1	1	1	0	1	1	1	0

Representation following permutation

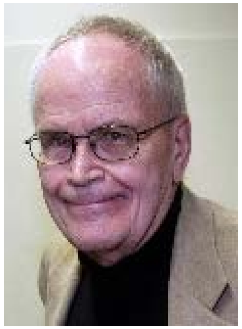
	I	Q	O	B	A	G	R	J	H	N	M	D	S	T	E	P	K	F	C	L
I	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0
J	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0
H	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	0	0	0
N	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0
M	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0
D	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0	0	0
S	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0
T	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0
E	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
P	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
K	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
C	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
L	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

- $N = 20$ subsystems and $E = 251$ *directed* interfaces; relatively dense ($\Delta = 0.661$)
- Girvan-Newman (2002) identifies 6 architectural communities with a modularity of just 0.017

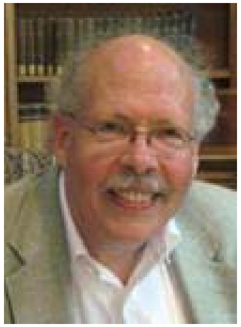
Girvan-Newman misses the indisputable, hierarchical clustering of subsystems!

. . . but blockmodeling does not.

- Blockmodeling
  - Partitions a network consisting of $i = 1, \cdots, N$ objects (i.e., the SV-3) into $k = 1, \cdots, P$ non-overlapping positions, where the positions generally abide the structure represented in a $(P \times P)$ image matrix such that $P \ll N$
  - Developed by computational sociologists at Harvard in the mid-1970's



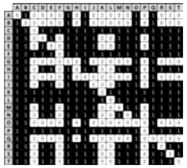
Harrison White



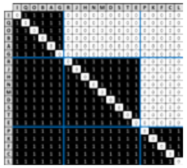
Scott Boorman

Ronald Breiger

Social structure from multiple networks. I. Blockmodels of roles and positions. *AJS*, 81(4), 730-780.



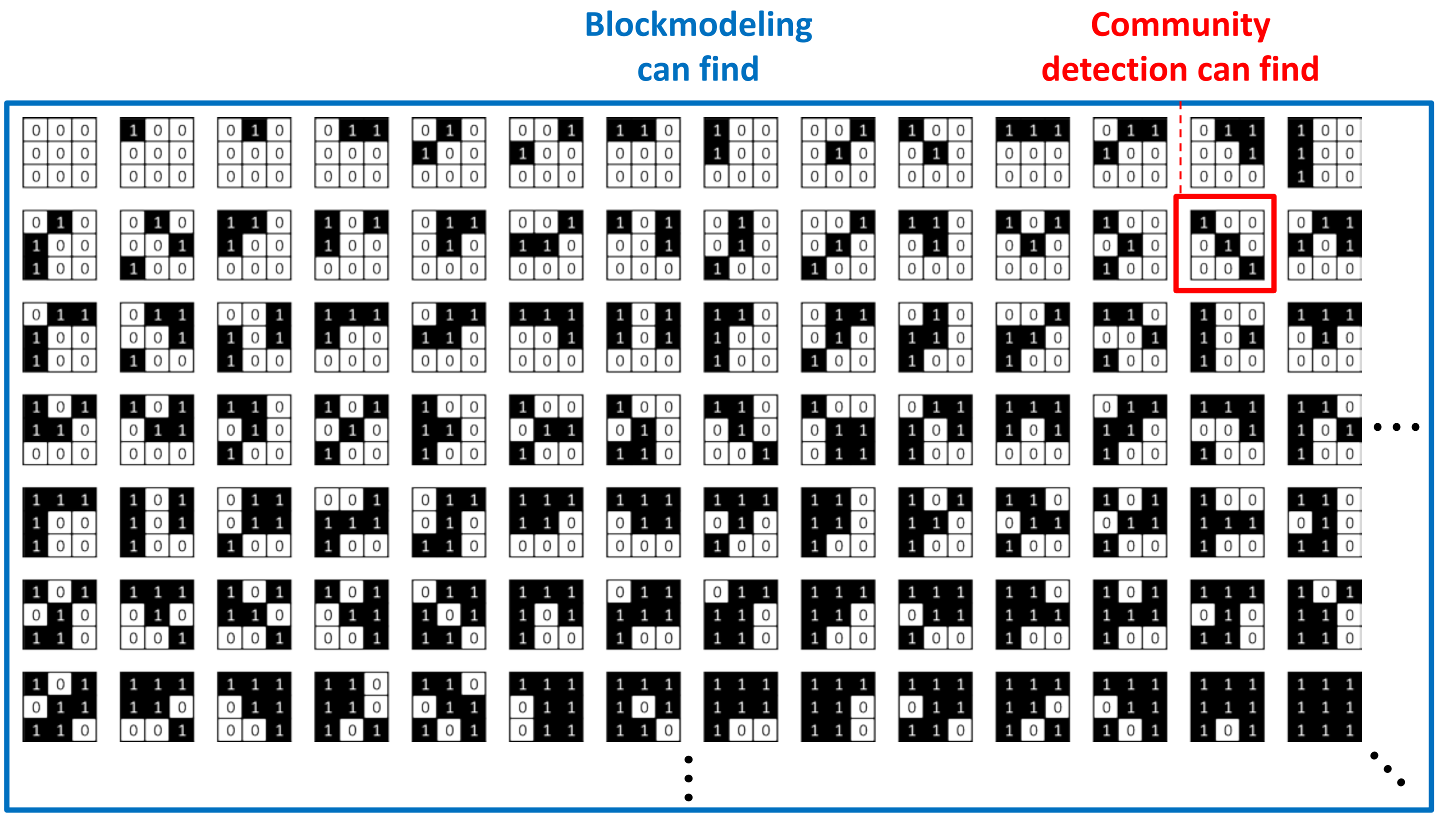
	B1	B2	B3
B1	1	0	0
B2	1	1	0
B3	1	1	1



Pajek recovers the SV-3's hierarchical structure exactly!

  - Integrated into popular network analysis software (i.e., Pajek via Doreian, Batagelj, and Ferligoj's (2005) direct approach)

Blockmodeling is the natural generalization of community detection . . .



Of the 512 possible (3×3) binary image matrices, community detection can find a partition for 1 – the identity; blockmodeling can accommodate all 512!

. . . but blockmodeling is not a panacea.

- Issue #1: Blockmodeling (BM) problems are NP-hard ⇒ time to find globally optimal solutions can *explode* as the # of subsystems/positions ↑

  Consequence #1: BM normally applies heuristics versus exact methods ⇒ better fitting image matrices and partitions may exist

- Issue #2: Exact methods largely confined to *confirmatory* fitting (image matrix is pre-specified) ⇒ exact *exploratory* fitting procedures are lacking

  Consequence #2: An SV-3's macrostructure is not "known in advance" ⇒ available exact BM methods are ill-suited for the task at hand

- Issue #3: Majority of BM heuristics and all exact methods focus on single one-/two-mode networks ⇒ BM multiple relations is an open problem

  Consequence #3: SV-3s are often mixed-mode networks ⇒ new methods are required to accommodate all SV-3s

WANTED . . . An Efficient Exact Method for Blockmodeling Mixed-Mode SV-3s

Given:

A mixed-mode SV-3

1-mode portion | **2-mode portion**

Internal Subsystems		I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	E1	E2	E3	E4	E5	E6	E7
		Internal Subsystems										External Subsystems						
	I1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	I2	1	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0
	I3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	I4	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	I5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	I6	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
	I7	0	1	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0
	I8	0	0	0	0	0	1	1	0	1	1	0	1	0	0	0	0	0
	I9	0	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1	1
	I10	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1

Find:

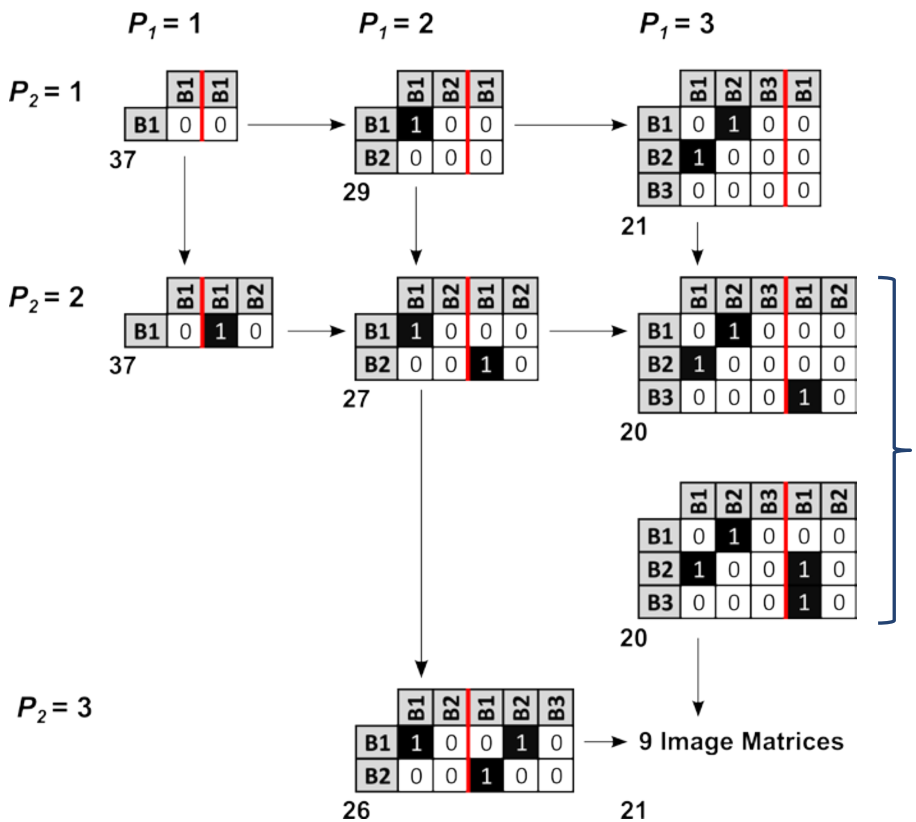
The globally optimal mixed-mode image matrix and corresponding partition with three or fewer internal and external subsystem positions

Internal Subsystems		B1	B2	B3	B1	B2	B3
		Internal Subsystems			External Subsystems		
	B1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1
	B2	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1
	B3	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1

Idea: Leverage the results in Brusco and Steinley's (2009) paper "Integer programs for one- and two-mode blockmodeling based on prespecified image matrices for structural and regular equivalence"

Globally Optimal IMs and Partition

- Formulated a series of IPs using C++; solved in IBM's ILOG CPLEX



		Internal Subsystems										External Subsystems						
		I1	I6	I7	I9	I10	I2	I8	I3	I4	I5	E2	E1	E3	E4	E5	E6	E7
Internal Subsystems	I1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
	I6	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0
	I7	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	0
	I9	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	1	1
	I10	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	1	1
	I2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	I8	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0
	I3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	I4	1	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0
	I5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Partition of internal subsystems driven by "outside" interfaces

Generalizing Dabkowski et al. (2014) via Blockmodeling

For a specified number of iterations . . .

Preprocessing

1. Initialize the system as the current system

2. Build an optimal set of $\{P(M = m), \beta\}$ pairs

3. Use Dabkowski-Fan-Breiger (2015; 2016) to identify an optimal P-position image matrix and partition of subsystems

Growth

4. Randomly select a member from the optimal set of $\{P(M = m), \beta\}$ pairs

5. Generate a realization for the incoming subsystem's (**X**'s) number of interfaces using $P(M = m)$; **if the IM and partition suggest a compelling, underlying architectural structure**, use *Connection Option A*; otherwise, use *Connection Option B*

Connection Option A (use blockmodel)

6a. Randomly assign **X** to position k,

6b. Model assignment of **X**'s m interfaces to positions as a random $(1 \times P)$ vector $\boldsymbol{C}$, where $\boldsymbol{C}$ follows a Multinomial$(m, \boldsymbol{p})$ distribution and $\boldsymbol{p}$ is the $(1 \times P)$ vector of multinomial probabilities given by

$$\frac{\text{\# interfaces in block } (k, l) \text{ of the partitioned, permuted SV−3}}{\text{\# interfaces in blocks } (k, \bullet) \text{ of the partitioned, permuted SV−3}};$$

generate a feasible realization for $\boldsymbol{C}$

6c. For $l = 1, \cdots, P$, attach **X** to $\boldsymbol{c}_l$ subsystems inside position l using attachment probabilities $p_i = {d_i}^{\beta} / \sum_{j=1}^{N} {d_j}^{\beta}$

6d. For each interface established in (6c), assign complexity ($w_{i\mathbf{X}}$)

Connection Option B (do not use blockmodel)

6a. Attach **X** to m subsystems using attachment probabilities $p_i = {d_i}^{\beta} / \sum_{j=1}^{N} {d_j}^{\beta}$

6b. For each interface established in (6a), assign complexity ($w_{i\mathbf{X}}$)

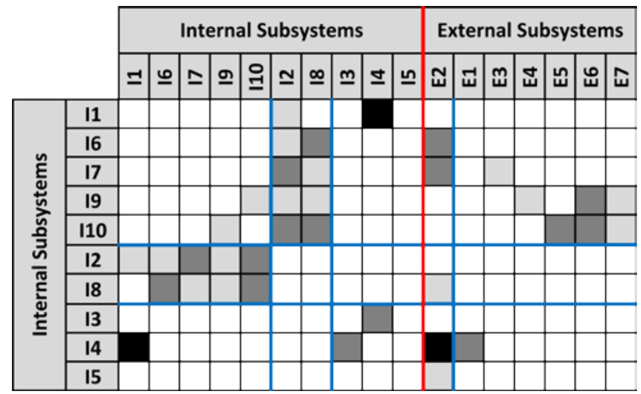
Cost Estimation (same as Dabkowski et. al (2014), goto Step 4)

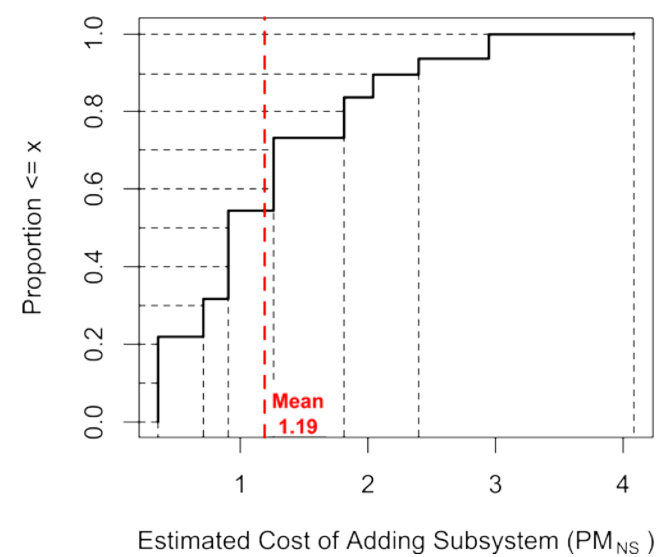
Estimating the Cost of Architectural Growth

- Assumed the following:
  - $A = 0.25; E = 1.06; \prod_{j=1}^{14} EM_j = 0.89$
  - Requirements: 75 easy, 50 nominal, 10 difficult
  - Internal interfaces: 6 easy, 5 nominal, 1 difficult
  - External interfaces: 6 easy, 6 nominal, 1 difficult

  $\Rightarrow$ 59.24 PM_{NS} of SE effort required
- Coded algorithm in R
- Ran 10,000 iterations

Expected cost to connect an additional subsystem to the SV-3's internal subsystems is (1.177, 1.206) PM_{NS}





Future Work

- Gather additional data for further validation and refinement
  - Secure sponsored research to weight SV-3s by interface complexity
  - Work with PMs to obtain multiple snapshots of SV-3s over time
- Explore additional connection options (e.g., model the probability that subsystem **X** is assigned to position *k* as a function of position *k*’s size)
- Modify algorithm to address external architectural growth
- Investigate the evolution of non-DoD architectures (e.g., open-source software architectures, non-militarized space systems, etc.)

Questions

CORRESPONDING AUTHOR:
LTC Matt Dabkowski
matthew.dabkowski@gmail.com
matthew.f.dabkowski.mil@mail.mil

PRESENTATION TITLE: Blockmodeling and the Estimation of Evolutionary Architectural Growth in Major Defense Acquisition Programs

IP formulation

For all possible mixed-mode image matrices solve . . .

Data

$\boldsymbol{S}^{(1)}$: $(N_1 \times N_1)$ 1-mode portion of the SV-3

$\boldsymbol{S}^{(2)}$: $(N_1 \times N_2)$ 2-mode portion of the SV-3

$\left(\boldsymbol{B}_1^{(r)} | \boldsymbol{B}_2^{(q)}\right)$: Binary $(P_1 \times P_1 | P_1 \times P_2)$ mixed-mode image matrix (IM), where $\boldsymbol{B}_1^{(r)}$ and $\boldsymbol{B}_2^{(q)}$ are the 1- and 2-mode portions

Indices

$i, j \in \mathcal{N}_1 \quad m \in \mathcal{N}_2$

$p, l \in \mathcal{P}_1 \quad k \in \mathcal{P}_2$

Decision variables

When using image matrix $\left(\boldsymbol{B}_1^{(r)} | \boldsymbol{B}_2^{(q)}\right)$:

$x_{i,p}^{(r,q)} = 1 \Rightarrow$ internal subsystem i assigned to internal position p

$y_{m,k}^{(r,q)} = 1 \Rightarrow$ external subsystem m assigned to external position k

Inconsistencies between one-mode IM and partition of internal subsystems | Inconsistencies between two-mode IM and partitions of internal and external subsystems

$$\min \underbrace{\sum_{p,l=1}^{P_1} \sum_{\substack{i,j=1 \\ i \neq j}}^{N_1} w_{i,j,p,l}^{(r,q)} \left(b_{p,l}^{(r)} + s_{i,j}^{(1)} - 2 b_{p,l}^{(r)} s_{i,j}^{(1)}\right)} + \underbrace{\sum_{i=1}^{N_1} \sum_{m=1}^{N_2} \sum_{p=1}^{P_1} \sum_{k=1}^{P_2} z_{i,m,p,k}^{(r,q)} \left(b_{p,k}^{(q)} + s_{i,m}^{(2)} - 2 b_{p,k}^{(q)} s_{i,m}^{(2)}\right)}$$

s.t.

$$\sum_{p=1}^{P_1} x_{i,p}^{(r,q)} = 1, \qquad \forall\, i \in \mathcal{N}_1$$

$$\sum_{k=1}^{P_2} y_{m,k}^{(r,q)} = 1, \qquad \forall\, m \in \mathcal{N}_2$$

Each internal / external subsystem assigned to a single internal / external position

$$\sum_{i=1}^{N_1} x_{i,p}^{(r,q)} \geq 1, \qquad \forall\, p \in \mathcal{P}_1$$

$$\sum_{m=1}^{N_2} y_{m,k}^{(r,q)} \geq 1, \qquad \forall\, k \in \mathcal{P}_2$$

No “empty” internal / external positions

$$x_{i,p}^{(r,q)} \in \{0,1\}, \qquad \forall\, i \in \mathcal{N}_1, p \in \mathcal{P}_1$$

$$y_{m,k}^{(r,q)} \in \{0,1\}, \qquad \forall\, m \in \mathcal{N}_2, k \in \mathcal{P}_2$$

Restrict decision variables to be binary

$$w_{i,j,p,l}^{(r,q)} \leq x_{i,p}^{(r,q)}, \qquad \forall\, i,j \in \mathcal{N}_1, p,l \in \mathcal{P}_1$$

$$w_{i,j,p,l}^{(r,q)} \leq x_{j,l}^{(r,q)}, \qquad \forall\, i,j \in \mathcal{N}_1, p,l \in \mathcal{P}_1$$

$$w_{i,j,p,l}^{(r,q)} \geq x_{i,p}^{(r,q)} + x_{j,l}^{(r,q)} - 1, \qquad \forall\, i,j \in \mathcal{N}_1, p,l \in \mathcal{P}_1$$

$$w_{i,j,p,l}^{(r,q)} \geq 0, \qquad \forall\, i,j \in \mathcal{N}_1, p,l \in \mathcal{P}_1$$

Linearization constraints for $w_{i,j,p,l}^{(r,q)} = x_{i,p}^{(r,q)} x_{j,l}^{(r,q)}$

$$z_{i,m,p,k}^{(r,q)} \leq x_{i,p}^{(r,q)}, \qquad \forall\, i \in \mathcal{N}_1, m \in \mathcal{N}_2, p \in \mathcal{P}_1, k \in \mathcal{P}_2$$

$$z_{i,m,p,k}^{(r,q)} \leq y_{m,k}^{(r,q)}, \qquad \forall\, i \in \mathcal{N}_1, m \in \mathcal{N}_2, p \in \mathcal{P}_1, k \in \mathcal{P}_2$$

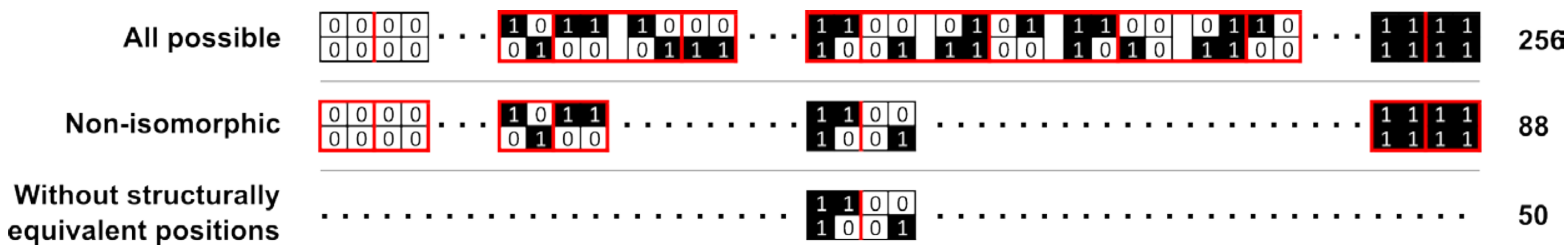
$$z_{i,m,p,k}^{(r,q)} \geq x_{i,p}^{(r,q)} + y_{m,k}^{(r,q)} - 1, \qquad \forall\, i \in \mathcal{N}_1, m \in \mathcal{N}_2, p \in \mathcal{P}_1, k \in \mathcal{P}_2$$

$$z_{i,m,p,k}^{(r,q)} \geq 0, \qquad \forall\, i \in \mathcal{N}_1, m \in \mathcal{N}_2, p \in \mathcal{P}_1, k \in \mathcal{P}_2.$$

Linearization constraints for $z_{i,m,p,k}^{(r,q)} = x_{i,p}^{(r,q)} y_{m,k}^{(r,q)}$

Two crucial structural observations

- <u>Observation #1</u>: Some image matrices are created "equal"
- <u>Observation #2</u>: Some positions are created "equal"



# of positions (P_1, P_2)	All possible IMs	Non-isomorphic IMs	IMs without structurally equivalent positions	Percentage of all possible IMs to fit
(1, 1)	4	4	4	100.00%
(1, 2)	8	6	2	25.00%
(2, 1)	64	36	32	50.00%
(2, 2)	256	88	50	19.53%
(2, 3)	1024	172	36	3.52%
(3, 1)	4096	752	688	16.80%
(3, 2)	32768	3272	2424	7.40%
(3, 3)	262144	10704	4912	1.87%
Total IMs to fit	300364	15034	8148	2.71%

Two order of magnitude reduction in the number of IMs to fit